Document made available under the Patent Cooperation Treaty (PCT)

International application number: PCT/US05/006453

International filing date: 01 March 2005 (01.03.2005)

Document type: Certified copy of priority document

Document details: Country/Office: US
Number: 60/551,417
Filing date: 09 March 2004 (09.03.2004)

Date of receipt at the International Bureau: 31 March 2005 (31.03.2005)

Remark: Priority document submitted or transmitted to the International Bureau in compliance with Rule 17.1(a) or (b)

1298914

THE UNITED STATES OF AMERICA

TO ALL TO WHOM THESE PRESENTS SHALL COME:

UNITED STATES DEPARTMENT OF COMMERCE

United States Patent and Trademark Office

March 21, 2005

THIS IS TO CERTIFY THAT ANNEXED HERETO IS A TRUE COPY FROM THE RECORDS OF THE UNITED STATES PATENT AND TRADEMARK OFFICE OF THOSE PAPERS OF THE BELOW IDENTIFIED PATENT APPLICATION THAT MET THE REQUIREMENTS TO BE GRANTED A FILING DATE.

APPLICATION NUMBER: ***60/551,417***
FILING DATE: ***March 09, 2004***
RELATED PCT APPLICATION NUMBER: ***PCT/US05/06453***

UNITED STATES PATENT AND TRADEMARK OFFICE
DEPARTMENT OF COMMERCE

Certified by

Jon W. Dudas

Under Secretary of Commerce
for Intellectual Property
and Director of the United States
Patent and Trademark Office

Please type a plus sign (+) inside this box → [+]   Express Mail Label:EV441452451US

PTO/SB/16 (8-00)
Approved for use through 10/31/2002. OMB 0651-0032
Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

22764 U.S. PTO 030904

22264 U.S. PTO 60/551417 030904

PROVISIONAL APPLICATION FOR PATENT COVER SHEET

This is a request for filing a PROVISIONAL APPLICATION FOR PATENT under 37 CFR 1.53 (c).

INVENTOR(S)		
Given Name (first and middle [if any])	Family Name or Surname	Residence (City and either State or Foreign Country)
Alexandros	Tourapis	West Windsor, New Jersey
Jill MacDonald	Boyce	Manalapan, New Jersey

☐ *Additional inventors are being named on the _____ separately numbered sheets attached hereto*

TITLE OF THE INVENTION (280 characters max)

REDUCED RESOLUTION SLICE UPDATE MODE FOR ADVANCED VIDEO CODING

CORRESPONDENCE ADDRESS

Direct all correspondence to:

☐ Customer Number [ ] → *Place Customer Number Bar Code Label here*

OR *Type Customer Number here*

☒ Firm *or* Individual Name	Joseph S. Tripoli - Thomson Licensing Inc.				
Address	PATENT OPERATIONS				
Address	P. O. BOX 5312				
City	PRINCETON	State	NJ	ZIP	08543-5312
Country	USA	Telephone	609-734-6834	Fax	609-734-6888

ENCLOSED APPLICATION PARTS ***(check all that apply)***

☒ Specification *Number of Pages* [12]   ☐ CD(s), Number [ ]

☒ Drawing(s) *Number of Sheets* [7]   ☐ Other (specify) [ ]

☐ Application Data Sheet. See 37 CFR 1.76

METHOD OF PAYMENT OF FILING FEES FOR THIS PROVISIONAL APPLICATION FOR PATENT *(check one)*

☐ Applicant claims small entity status. See 37 CFR 1.27.

☐ A check or money order is enclosed to cover the filing fees

☒ The Commissioner is hereby authorized to charge filing fees or credit any overpayment to Deposit Account Number: [07-0832]   FILING FEE AMOUNT ($) [160]

☐ Payment by credit card. Form PTO-2038 is attached.

The invention was made by an agency of the United States Government or under a contract with an agency of the United States Government.

☒ No.

☐ Yes, the name of the U.S. Government agency and the Government contract number are: _____.

Respectfully submitted,
SIGNATURE [signature]   Date [03/09/04]

TYPED or PRINTED NAME GUY H. ERIKSEN   REGISTRATION NO. (*if appropriate*) [41,736]

TELEPHONE (609) 734-6807   Docket Number: [PU040073]

USE ONLY FOR FILING A PROVISIONAL APPLICATION FOR PATENT

Express Mail Label: EV441452451US

REDUCED RESOLUTION SLICE UPDATE MODE FOR ADVANCED VIDEO CODING

The invention extends the Reduced Resolution Update Mode, currently supported by the H.263, into the new H.264 (MPEG-4 AVC,/JVT) video coding standard. This mode provides the opportunity to increase the coding picture rate, while maintaining sufficient subjective quality. This is done by encoding an image at a reduced resolution, while performing prediction using a high resolution reference. This allows the final image to be reconstructed at full resolution and with good quality, although the bitrate required to encode the image has been reduced considerably. Considering that H.264 contains several new tools and concepts compared to its counterpart, this concept had to be modified to fit within the specifications of the new standard or its extensions. This includes new syntax elements, and certain semantic and encoder/decoder architecture modifications to inter and intra prediction modes. Impact on other tools supported by the H.264 standard, such as Macroblock Based Adaptive Field/Frame mode, are also presented.

The H.264 (or JVT, or MPEG-4 AVC) standard has introduced several new features that allow it to achieve considerable coding efficiency improvement compared to older standards such as MPEG-2/4, and H.263. Nevertheless, although H.264 contains most of the algorithmic features of older standards, some were never ported. One of these features was the consideration of the Reduced-Resolution Update mode that already exists within H.263. This mode provides the opportunity to increase the coding picture rate, while maintaining sufficient subjective quality. This is done by encoding an image at a reduced resolution, while performing prediction using a high resolution reference, which allows also the final image to be reconstructed at full resolution. This mode was found useful in H.263 especially during the presence of heavy motion within the sequence since it allowed an encoder to maintain high frame rate (and thus improved temporal resolution) while also maintaining high resolution and quality in stationary areas.

The *Reduced-Resolution Update* mode was introduced in H.263 to allow an increase in the coding picture rate while maintaining sufficient subjective quality. Although the syntax of a bitstream encoded in this mode was essentially identical to

a bitstream coded in full resolution, the main difference was on how all modes within the bitstream were interpreted, and how the residual information was considered and added after motion compensation. More specifically, an image in this mode had ¼ the number of macroblocks compared to a full resolution coded picture, while motion vector data was associated with block sizes of 32×32 and 16×16 of the full resolution picture instead of 16×16 and 8×8 respectively. On the other hand, DCT and texture data are associated with 8x8 blocks of a reduced resolution image, while an upsampling process is required in order to generate the final full image representation.

Although this process could result in reduction in objective quality, this is more than compensated from the reduction of bits that need to be encoded due to the reduced number (by 4) of modes, motion data, and residuals. This is especially important at very low bitrates where modes and motion data can be considerably more than the residual. Subjective quality was also far less impaired compared to objective quality. Also, this process can be seen somewhat similar to the application of a low pass filter on the residual data prior to encoding, which, however, requires the transmission of all modes, motion data, and filtered residuals, thus being less efficient.

This concept was never introduced within H.264.

The *Reduced-Resolution Update* (*RRU*) mode can be ported into H.264 and extended. Certain aspects of the codec need to be now considered with regards to this new mode. More specifically, it is necessary to introduce a new slice parameter (**reduced_resolution_update**) according to which the current slice is subdivided into (RRUwidth * 16) × (RRUheight * 16) size macroblocks. Unlike in H.263, it is not necessary that RRUwidth be equal to RRUheight. Additional slice parameters can be included, more specifically **rru_width_scale** = RRUwidth and **rru_height_scale** = RRUheight which allow us to reduce resolution horizontally or vertically at any ratio we may desire (Table 2). Possible options, for example, include scaling by 1 horizontally & 2 vertically (MBs are of size 16x32), 2 vertically & 1 horizontally (MB size 32x16), or in general have MBs of size (rru_width_scale*16)x(rru_height_scale*16).

Without loss in generality, we discuss the case where RRUwidth = RRUheight = 2 and the macroblocks are of size 32×32. In this case, all

macroblock partitions and sub-partitions have to be scaled by 2 horizontally and 2 vertically (Figure 1). Unlike H.263 where motion vector data had to be divided by 2 to conform to the standards specifics, this is not necessary in H.264 and motion vector data can be coded in full resolution/subpel accuracy. Skipped macroblocks in P slices are in this mode considered as of having 32×32 size, while the process for computing their associated motion data remains unchanged, although we need to now consider 32x32 neighbors instead of 16x16.

Another key difference of this invention, although optional, is that in H.264 texture data do not have to represent information from a lower resolution image. Since intra coding in H.264 is performed through the consideration of spatial prediction methods using either 4×4 or 16×16 block sizes, this can be extended, similarly to inter prediction modes, to 8×8 and 32×32 intra prediction block sizes. Prediction modes nevertheless remain more or less the same, although now more samples are used to generate the prediction signal (Figure 2). For example, for 8×8 vertical prediction we now use samples C0-C7, while DC prediction is the mean of C0-C7 and R0-R7. Furthermore, all diagonal predictions need to also consider samples C8-C15. A similar extension can be applied to the 32×32 intra prediction mode.

The residual data is then downsampled and is coded using the same transform and quantization process already available in H.264. The same process is applied for both Luma and Chroma samples. During decoding the residual data needs to be upsampled. The downsampling process is done only in the encoder, and hence does not need to be standardized. The upsampling process must be matched in the encoder and the decoder, and so must be standardized. Possible upsampling methods that could be used are the zero or first order hold or by considering a similar strategy as in H.263 (Figure 3).

H.264 also considers an in-loop deblocking filter, applied to 4×4 block edges. Since currently the prediction process is now applied to block sizes of 8×8 and above, we also modify this process to consider 8×8 block edges instead.

Different slices in the same picture may have different values of **reduced_resolution_update, rru_width_scale** and **rru_height_scale**. Because the in-loop deblocking filter is applied across slice boundaries, blocks on either side of the slice boundary may have been coded at different resolutions. In this case we

need to consider for the deblocking filter parameters computation, the largest QP value among the two neighboring 4x4 normal blocks on a given 8x8 edge, while the strength of the deblocking is now based on the total number of non zero coefficients of the two blocks.

To support Flexible Macroblock Ordering as indicated by **num_slice_groups_minus1** greater than 0 in the picture parameter sets, with Reduced Resolution Update mode, it is also necessary to transmit in the picture parameter set an additional parameter named as **reduced_resolution_update_enable** (Table 1). It is not allowed to encode a slice using the Reduced Resolution Mode if FMO is present and this parameter is not set. Furthermore if this parameter is set, we need to also transmit the parameters **rru_max_width_scale** and **rru_max_height_scale**. These parameters are necessary to ensure that the map provided can always support the current Reduced Resolution macroblock size. This means that it is necessary for these parameters to conform to the following conditions:

max_width_scale % rru_width_scale=0,
max_height_scale % rru_height_scale=0 and,
max_width_scale>0, max_height_scale>0.

The FMO slice group map that is transmitted corresponds to the lowest allowed reduced resolution, corresponding to **rru_max_width_scale** and **rru_max_height_scale**. Note that if multiple macroblock resolutions are used then **rru_max_width_scale** and **rru_max_height_scale** need to be multiples of the *least common multiple* of all possible resolutions within the same picture.

Direct modes in H.264 are affected depending on whether the current slice is in reduced resolution mode, or the list1 reference is in reduced resolution mode and the current one is not. For the direct mode case, when the current picture is in reduced resolution and the reference picture is of full resolution, we borrow from a similar method currently employed within H.264 when *direct_8x8_inference_flag* is enabled. According to this method, co-located partitions are assigned by considering only the corresponding corner 4×4 blocks (corner is based on block indices) of an 8×8 partition. In our case, if direct belongs to a reduced resolution

slice, motion data for the co-located partition are derived as if *direct_8x8_inference_flag* was set to 1. This can be seen also as a downsampling of the motion field of the co-located reference. Although not necessary, if *direct_8x8_inference_flag* was already set within the bitstream, this process could be applied twice. This process can be seen more clearly in Figure 4. For the case when the current slice is not in reduced resolution mode, but its first list1 reference is, it is necessary to first upsample all motion data of this reduced resolution reference. Motion data can be upsampled using zero order hold, which is the method with the least complexity. Other filtering methods, for example similar to the process used for the upsampling of the residual data, could also be used.

Some other tools of H.264 are also affected through the consideration of this mode. More specifically, macroblock adaptive field frame mode (MB-AFF) needs to be now considered using a 32×64 super-macroblock structure. The upsampling process is performed on individual coded block residuals. If field pictures are coded the blocks are coded as field residuals, and hence the upsampling is done in fields. Similarly, when MB-AFF is used individual blocks are coded either in field of frame mode, and their corresponding residuals are upsampled in field or frame mode respectively.

To allow the reduced resolution mode to work for all possible resolutions, a picture is always extended vertically and horizontally in order to be always divisible by 16 * **rru_height_scale** and 16 * **rru_width_scale**, respectively. For the example where rru_height_scale = rru_width_scale = 2, the original resolution of an image was $H_R \times V_R$ the image is padded to a resolution equal to $H_C \times V_C$ where:

$$H_C = ((H_R + 31) / 32) * 32$$
$$V_C = ((V_R + 31) / 32) * 32$$

The process for extending the image resolution is similar to what is currently done for H.264 (Figure 5) to extend the picture size to be divisible by 16.

The extended luminance for a QCIF resolution picture is given by the following formula:

$R_{RRU}(x, y) = R(x', y')$,

where

x, y = spatial coordinates of the extended referenced picture in the Pixel domain,

x', y' = spatial coordinates of the referenced picture in the pixel domain,

$R_{RRU}(x, y)$ = pixel value of the extended referenced picture at (x, y),

$R(x', y')$ = pixel value of the referenced picture at (x', y'),

$$x' = 175 \text{ if } x > 175 \text{ and } x < 192$$
$$= x \text{ otherwise,}$$

$$y' = 143 \text{ if } y > 143 \text{ and } y < 160$$
$$= y \text{ otherwise,}$$

A similar approach is used for extending chroma samples, but to half of the size.

A prototype encoder is shown in Figure 6 while a simplified decoder model is shown in Figure 7. This model can be extended and improved by using additional processing elements, such as spatio-temporal analysis in both the encoder and decoder, which would allow us to remove some of the artifacts introduced through the residual downsampling and upsampling process.

A variation of the above approach is to allow the use of reduced resolutions not just at the slice level, but also at the macroblock level. Although we may have different variations of this approach, one approach is to signal resolution variation through the usage of the reference picture indicator. Reference pictures could be associated implicitly (i.e. odd/even references) or explicitly (through a transmitted table in the slice parameters) with the transmission of full or reduced resolution residual. If a 32×32 macroblock is coded using reduced residual, then a single *coded block pattern (cbp)* is transmitted associated with the transform coefficients of the 16 reduced resolution blocks. Otherwise, we need to transmit 4 cbp (or a single combined one), which are associated with 64 full resolution blocks. Note that for this method to work, all blocks within this macroblock need to be coded in the same

resolution. This method requires the transmission of an additional table, which would provide the information regarding the scaling, or not of the current reference, including the scaling parameters, similarly to what is currently done for weighted prediction.

Anticipated/Sample Claims and Claimed Elements:

1. A video encoder that applies a downsampling operation to a slice's prediction residual prior to block transform and quantization.
2. Claim 1 where the downsampling operation may be different for the horizontal and vertical directions (and may be applied in only one of the two directions).
3. Claim 2 where the downsampling resolution is signaled by parameters in the coded slice.
4. Claim 1, where the residual signal is formed after intra prediction.
5. Claim 4 where intra prediction is performed using 8x8 or 32x32 prediction modes.
6. Claim 1, where the residual signal is formed after inter prediction.
7. Claim 4, where inter prediction is performed using 32x32 macroblocks, and 32x32, 32x16, 16x32, and 16x16 macroblock partitions, or 16x16,16x8, 8x16, and 8x8 sub-macroblock partitions.
8. Decoder that receives and decodes a stream complying with Claim 1, by upsampling the residual prior to adding it to the predicted reference.
9. Claim 1 where encoder allows reduced resolution update mode to be inferred at the macroblock level using reference indices.
10. Claim 5 where decoder can infer whether a macroblock is in reduced resolution update mode based on its reference indices, and decode it after upscaling if necessary it's associated residual.
11. Claim 1 where additional support for Flexible Macroblock ordering has been introduced.
12. Claim 1 where for interlace pictures downsampling/upsampling is performed in the mode that the current block/macroblock has been encoded (either field or frame).

Table 1. H.254 Picture parameter syntax with consideration of Reduced Resolution Update Mode.

pic_parameter_set_rbsp( ) {	C	Descriptor
pic_parameter_set_id	1	ue(v)
seq_parameter_set_id	1	ue(v)
entropy_coding_mode_flag	1	u(1)
pic_order_present_flag	1	u(1)
num_slice_groups_minus1	1	ue(v)
if(num_slice_groups_minus1 > 0) {		
/* Consideration of RRU */		
reduced_resolution_update_enable	1	u(1)
if(!reduced_resolution_update) {		
rru_max_width_scale	1	u(v)
rru_max_height_scale	1	u(v)
}		
/* End of Reduced Resolution Update Parameters */		
slice_group_map_type	1	ue(v)
if(slice_group_map_type = = 0)		
for(iGroup = 0; iGroup <= num_slice_groups_minus1; iGroup++)		
run_length_minus1[iGroup **]**	1	ue(v)
else if(slice_group_map_type = = 2)		
for(iGroup = 0; iGroup < num_slice_groups_minus1; iGroup++) {		
top_left[iGroup **]**	1	ue(v)
bottom_right[iGroup **]**	1	ue(v)
}		
else if(slice_group_map_type = = 3 || slice_group_map_type = = 4 || slice_group_map_type = = 5) {		
slice_group_change_direction_flag	1	u(1)
slice_group_change_rate_minus1	1	ue(v)
} else if(slice_group_map_type = = 6) {		

pic_size_in_map_units_minus1	1	ue(v)
for(i = 0; i <= pic_size_in_map_units_minus1; i++)		
slice_group_id[i]	1	u(v)
}		
}		
num_ref_idx_l0_active_minus1	1	ue(v)
num_ref_idx_l1_active_minus1	1	ue(v)
weighted_pred_flag	1	u(1)
weighted_bipred_idc	1	u(2)
pic_init_qp_minus26 /* relative to 26 */	1	se(v)
pic_init_qs_minus26 /* relative to 26 */	1	se(v)
chroma_qp_index_offset	1	se(v)
deblocking_filter_control_present_flag	1	u(1)
constrained_intra_pred_flag	1	u(1)
redundant_pic_cnt_present_flag	1	u(1)
rbsp_trailing_bits( )	1	
}		

Table 2. H.254 Slice header syntax with consideration of Reduced Resolution Update Mode.

slice_header( ) {	C	Descriptor
first_mb_in_slice	2	ue(v)
slice_type	2	ue(v)
pic_parameter_set_id	2	ue(v)
frame_num	2	u(v)
/* **Reduced Resolution Update parameters** */		
reduced_resolution_update	2	u(1)
/* Following is optional*/		
if(!reduced_resolution_update) {		
rru_width_scale	2	u(v)
rru_height_scale	2	u(v)
}		
/* End of Reduced Resolution Update Parameters */		
if(!frame_mbs_only_flag) {		
field_pic_flag	2	u(1)
if(field_pic_flag)		
bottom_field_flag	2	u(1)
}		
if(nal_unit_type = = 5)		
idr_pic_id	2	ue(v)
if(pic_order_cnt_type = = 0) {		
pic_order_cnt_lsb	2	u(v)
if(pic_order_present_flag && !field_pic_flag)		
delta_pic_order_cnt_bottom	2	se(v)
}		
if(pic_order_cnt_type = = 1 && !delta_pic_order_always_zero_flag) {		
delta_pic_order_cnt[0]	2	se(v)
if(pic_order_present_flag && !field_pic_flag)		
delta_pic_order_cnt[1]	2	se(v)

}		
if(redundant_pic_cnt_present_flag)		
redundant_pic_cnt	2	ue(v)
if(slice_type = = B)		
direct_spatial_mv_pred_flag	2	u(1)
if(slice_type = = P || slice_type = = SP || slice_type = = B) {		
num_ref_idx_active_override_flag	2	u(1)
if(num_ref_idx_active_override_flag) {		
num_ref_idx_l0_active_minus1	2	ue(v)
if(slice_type = = B)		
num_ref_idx_l1_active_minus1	2	ue(v)
}		
}		
ref_pic_list_reordering( )	2	
if((weighted_pred_flag && (slice_type = = P || slice_type = = SP)) || (weighted_bipred_idc = = 1 && slice_type = = B))		
pred_weight_table( )	2	
if(nal_ref_idc != 0)		
dec_ref_pic_marking( )	2	
if(entropy_coding_mode_flag && slice_type != I && slice_type != SI)		
cabac_init_idc	2	ue(v)
slice_qp_delta	2	se(v)
if(slice_type = = SP || slice_type = = SI) {		
if(slice_type = = SP)		
sp_for_switch_flag	2	u(1)
slice_qs_delta	2	se(v)
}		
if(deblocking_filter_control_present_flag) {		
disable_deblocking_filter_idc	2	ue(v)
if(disable_deblocking_filter_idc != 1) {		
slice_alpha_c0_offset_div2	2	se(v)

slice_beta_offset_div2	2	se(v)
}		
}		
if(num_slice_groups_minus1 > 0 && slice_group_map_type >= 3 && slice_group_map_type <= 5)		
slice_group_change_cycle	2	u(v)
}		

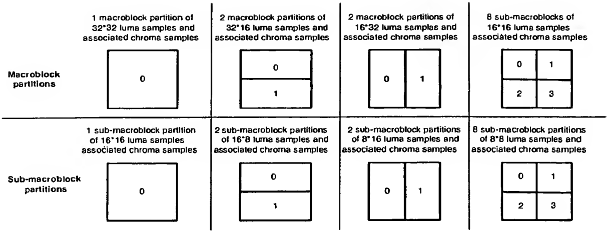
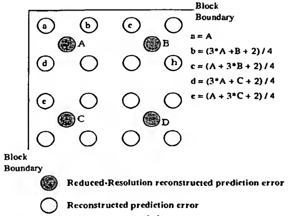


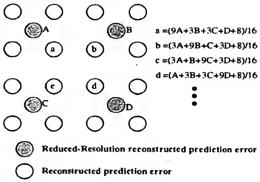
Figure 1. Macroblock and Sub-macroblock partitions in Reduced Resolution Update Mode

X	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15
R0	a00	a01	a02	a03	a04	a05	a06	a07								
R1	a10	a11	a12	a13	a14	a15	a16	a17								
R2	a20	a21	a22	a23	a24	a25	a26	a27								
R3	a30	a31	a32	a33	a34	a35	a36	a37								
R4	a40	a41	a42	a43	a44	a45	a46	a47								
R5	a50	a51	a52	a53	a54	a55	a56	a57								
R6	a60	a61	a62	a63	a64	a65	a66	a67								
R7	a70	a71	a72	a73	a74	a75	a76	a77								

Figure 2. Samples (C0-C15, X, and R0-R7) used for 8×8 intra prediction



(a)



(b)

Figure 3. Residual up-sampling process (a) for block boundaries, and b) for inner positions.

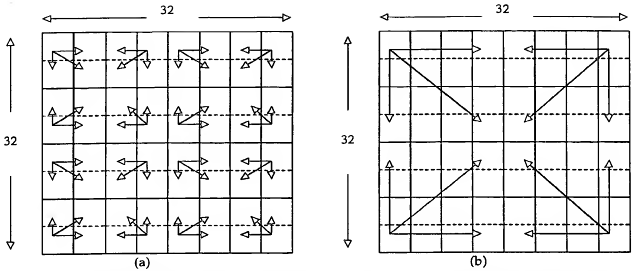


Figure 4. Motion inheritance for direct mode if current is in reduced resolution and first list1 reference is in full resolution when *direct_8x8_inference_flag* is set to (a) 0 and (b) 1.

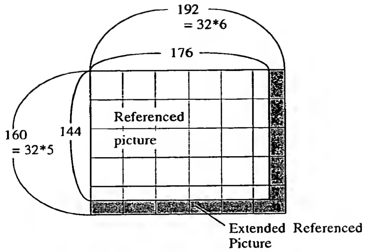


Figure 5. Resolution extension for a QCIF resolution picture

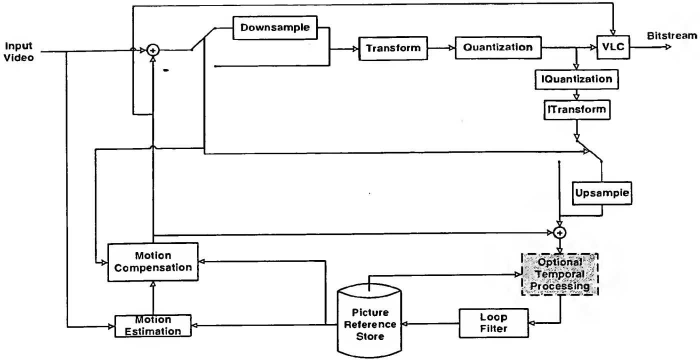


Figure 6. Prototype encoder supporting Reduced Resolution Update mode

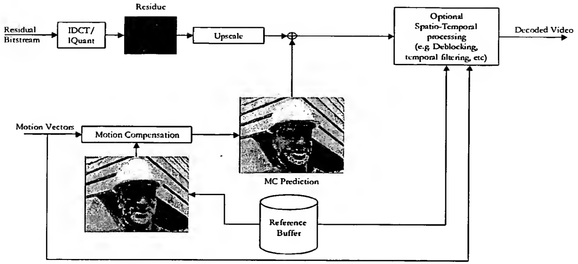


Figure 7. Simplified Decoder model.